# Project Design Phase-II

|  |  |
| --- | --- |
| **Date** | 25 June 2025 |
| **Team ID** | LTVIP2025TMID53966 |
| **Project Name** | BookNest: Where Stories Nestle |
| **Mentor Name** | Dr Shaik Salma Begam |
| **Maximum Marks** | 4 Marks |

**Technology Stack (Architecture & Stack)**

**Technical Architecture**

**BookNest** is a modern, MERN stack-based online book store built for scalability, performance, and user satisfaction. The architecture includes a **React-based frontend**, **Node.js/Express.js backend**, **MongoDB** for data storage, and integrates essential services like authentication, inventory, and order processing. The system is modular and ready for cloud deployment with future integrations like recommendation engines and secure payment gateways.

**Table 1: Technology Stack Components**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Component | Description | Technology / Service Used |
| 1 | User Interface | Web UI for readers to browse, search, and purchase books | React.js, HTML5, CSS3, JavaScript |
| 2 | Application Logic-1 | |  | | --- | | User registration, login, authentication, session management |  |  | | --- | |  | | Node.js, Express.js, JWT |
| 3 | Application Logic-2 | |  | | --- | | Book browsing, filtering, category selection, cart, checkout |  |  | | --- | |  | | Node.js, Express.js |
| 4 | Database | |  | | --- | | Data storage for users, books, orders, inventory |  |  | | --- | |  | | MongoDB, Mongoose |
| 5 | Cloud Database | |  | | --- | | (Optional) Cloud-hosted MongoDB for scalability |  |  | | --- | |  | | MongoDB Atlas |
| 6 | File Storage | |  | | --- | | Book cover images, author photos |  |  | | --- | |  | | Local filesystem, (optionally AWS S3) |
| 7 | External API-1 | |  | | --- | | (Optional) Payment gateway integration |  |  | | --- | |  | | |  | | --- | | Stripe API, Razorpay API |  |  | | --- | |  | |
| 8 | External API-2 | |  | | --- | | (Optional) Email notifications for orders |  |  | | --- | |  | | |  | | --- | | Nodemailer, SendGrid |  |  | | --- | |  | |
| 9 | |  | | --- | |  |  |  | | --- | | **Recommendation Engine** | | |  | | --- | | (Future) Personalized book suggestions |  |  | | --- | |  | | |  | | --- | | Python (Flask API), TensorFlow, Scikit-learn |  |  | | --- | |  | |
| 10 | Infrastructure | |  | | --- | | Deployment, containerization, and scaling |  |  | | --- | |  | | |  | | --- | | Docker, Heroku, AWS EC2, Nginx | |

**Table 2: Application Characteristics**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Characteristics | Description | Technology / Approach Used |
| 1 | Open-Source Frameworks | |  | | --- | | Built entirely on open-source tools for community support and rapid development |  |  | | --- | |  | | React.js, Node.js, Express.js, MongoDB |
| 2 | Security Implementations | |  | | --- | | Secure authentication and data protection mechanisms |  |  | | --- | |  | | JWT, Helmet, CORS, OWASP practices |
| 3 | Scalable Architecture | |  | | --- | | Modular services with REST APIs, cloud-deployable structure |  |  | | --- | |  | | Docker, AWS, MongoDB Atlas, Microservices (future) |
| 4 | Availability | |  | | --- | | High availability through cloud deployment and load balancing |  |  | | --- | |  | | |  | | --- | | AWS EC2, Heroku, MongoDB Atlas, Nginx |  |  | | --- | |  | |
| 5 | Performance | |  | | --- | | Optimized backend queries, fast API responses, and image delivery via CDN |  |  | | --- | |  | | MongoDB indexing, Redis (future), Cloudflare CDN |